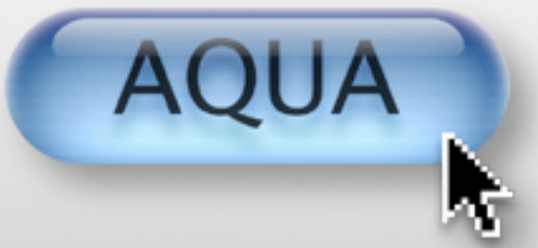
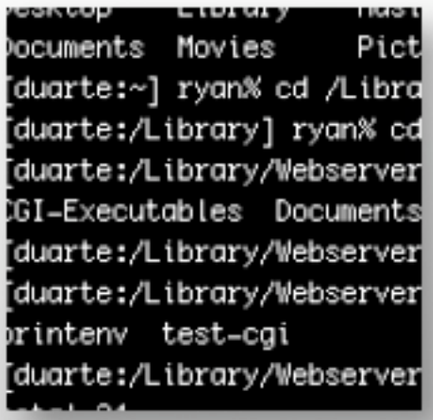
Understanding Safari Technology and Web Standards

Session 411





Understanding Safari Technology and Web Standards

Darin Adler
Manager, Safari Engineering
darin@apple.com

Safari Technology Matters To…

- Website developers
- Macintosh application developers
- Apple Help developers
- Developers interested in how web browsers work

Expect to Hear About…

- The Web Kit, Safari technology for your application
- The Web Kit's support for web standards
- The Web Kit's approach to website compatibility
- The Web Kit and open source
- Our goals for the future

Safari

Safari Success

- Public beta—released in January
- Millions of copies downloaded
- Safari 1.0—released this week

Why So Popular?

- Impressive performance
- Innovative features
- Solid quality

The Web Kit

What Is the Web Kit?

Safari's engine

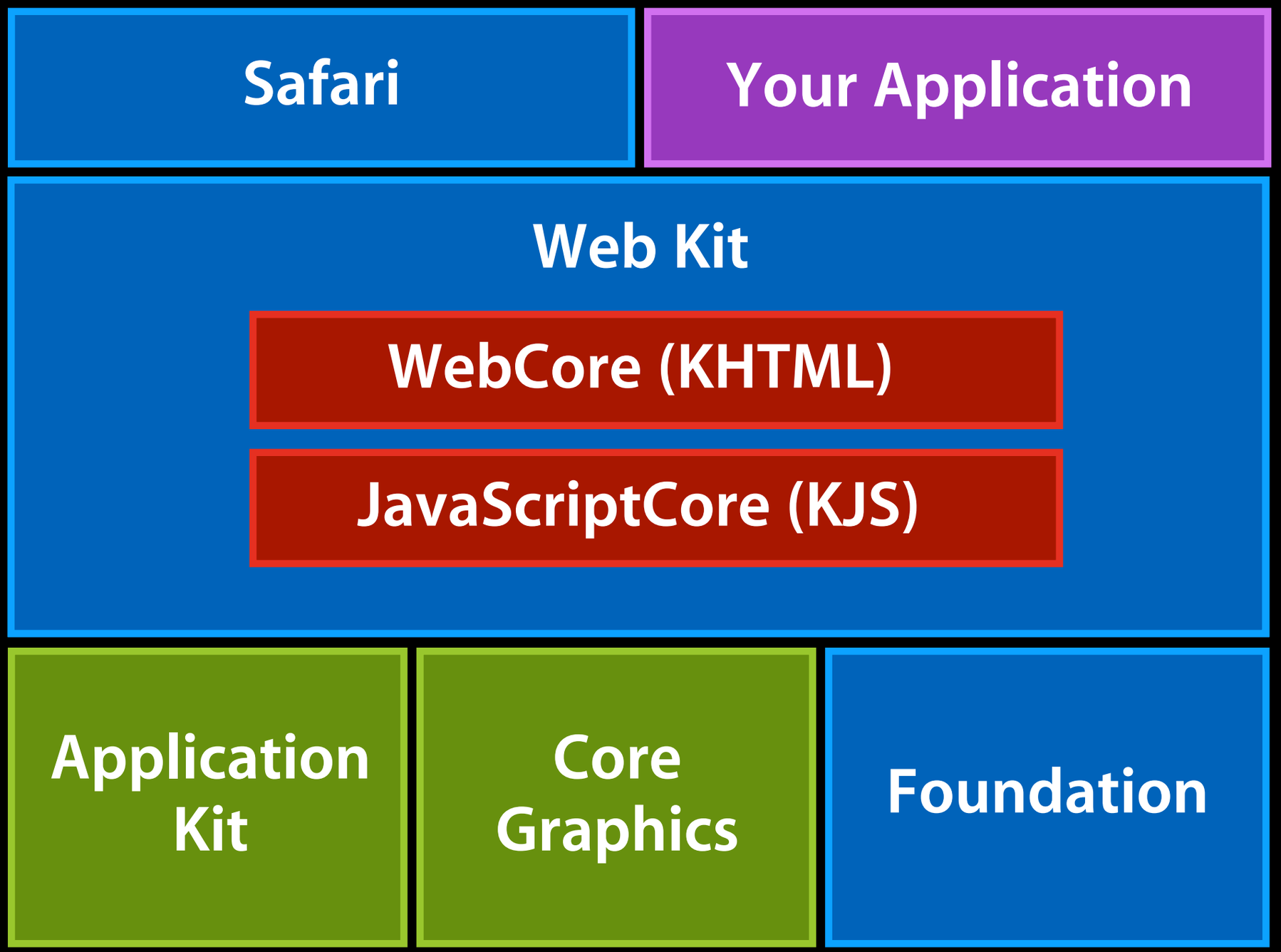
What Is the Web Kit?

- Safari's web technology engine
- A public framework
- Available for your application now

Can I Use the Web Kit?

- Yes, in Cocoa and Carbon applications
  - On 10.2, with Safari 1.0 installed
  - On Panther
- Objective-C API
- C API for putting a web view in a Carbon window

The Web Kit in Context



The Web Kit and Other Frameworks

- The Web Kit contains WebCore and JavaScriptCore
  - Completely encapsulates them
  - WebCore and JavaScriptCore have no API
- Foundation URL API
  - Used by the Web Kit, also directly by applications
- Safari uses the Web Kit and the Foundation URL API

Web Standards

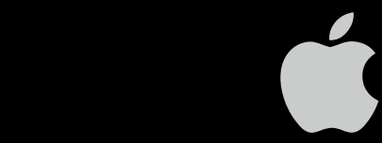
Some Design Goals of the Web Kit

- Make web pages work
- Make web pages look good
- Support standards

- Show pages the way the designer intended
  - Can’t just clone Windows Internet Explorer

Supported Standards

- HTML 4.01
- XHTML 1.1
- CSS 1, 2, and the emerging CSS 3
- ECMAScript 262 edition 3 (JavaScript 1.4)
- DOM levels 1 and 2
- DOM extensions
  - "DOM level 0"
  - Some Microsoft extensions, including innerHTML



More Supported Standards

- Netscape cookie de facto standard
- Netscape plug-in API
- Java applets
- Tons of different character sets, Unicode internally
- Bi-directional text and contextual letterforms

Even More Supported Standards

- HTTP 1.1
  - Chunked transfers
  - Persistent connections
- Basic and digest authentication
- SSL v3, TLS v1, for secure HTTP
- HTTP proxy servers
- SOCKS proxy servers

CSS

- A W3C standard
- Cascading Style Sheets
- Separate presentation from content
- Great layout without using tables

```
form { margin: 3px; padding: 0px; }
```

DOM

- A W3C standard
- Document Object Model
- Manipulate document from JavaScript
- Achieve dynamic effects

```
document.langForm.Japanese.checked = 1;
```

Plug-ins

- A Netscape de facto standard
- Documentation on Netscape's website
- Existing plug-ins work in any Web Kit application
- Mach-O for best performance
- Significant enhancements planned for the future

Java

- A Sun standard
- Mac OS X includes the latest released version of Suns Java Standard Edition, J2SE 1.4.1
- Plug-in for applets supports all Sun Java Plug-in Features
  - Java console for debugging
  - Certificates and caching integrated with the Web Kit

Examples

Website Compatibility

How to Create a Compliant Website

- Learn web standards
- Test your site with lots of browsers
- Test your site with Safari!
- Validate with tools like the ones from the W3C

A Bit of History

- Not too long ago, there was only one important browser
- Many sites depended on quirks in Netscape Navigator
- Then, Microsoft Internet Explorer showed up
- Tables, for example, had different quirks in each
- Many sites effectively had two versions

Things to Avoid

- The two browser trap
  - document.all means Internet Explorer
  - document.layers means Netscape Navigator 4
- Parsing the “user agent string”
- Requiring non-standard extensions
  - For example, createTextRange and caretPos
  - Often, there are standard alternatives

Browser-Independent JavaScript

- Website developers do need to test for capabilities, if using a feature that is not universally available

✅ Use object detection

  - If the object is not there, JavaScript returns “undefined”
  - Check before setting
  - Also works on individual methods

✅ Use the W3C standard way of finding objects

  - getElementByID(”elementName”)

JavaScript Debugging

- New for Safari 1.0: “Log JavaScript Exceptions”
- Exceptions will be logged to the Console
- Turn on the Safari Debug menu

```
defaults write com.apple.safari IncludeDebugMenu YES
```

- Turn the logging using an item in the Debug menu

Web Realities

- Real-world web pages still must work with the Web Kit
- Poor web pages have implicit assumptions
  - De facto standard is “what Windows IE does”
  - The browser will tolerate bad HTML
- Easy to make a bad page, everyone’s doing it!

The Web Kit's User Agent String

- Designed for compatibility, like all user agent strings!
- Look like other standards-compliant browsers
  - Even to poorly designed web pages
- Include a unique identifier
  - Say it's the Web Kit
  - Identify the Web Kit version
  - Name the application and give its version

Beyond the User Agent String

- Methods on the navigator object like navigator.platform
- Goals are the same
  - Maximize compatibility
  - Don’t hide identity

Sample User Agent String

Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-us) AppleWebKit/XX (KHTML, like Gecko) YourAppName/YY

-
- Every browser is Mozilla!
- XX is the version of the Web Kit
- YourAppName/YY is the application name and version

Sample Navigator Properties

appCodeName = Mozilla

appName = Netscape

appVersion = 5.0 (Macintosh; U; PPC Mac OS X; en-us)
AppleWebKit/XX (KHTML, like Gecko)
YourAppName/YY

platform = MacPPC

product = Gecko

productSub = 20030107

vendor = Apple Computer, Inc.

More Examples

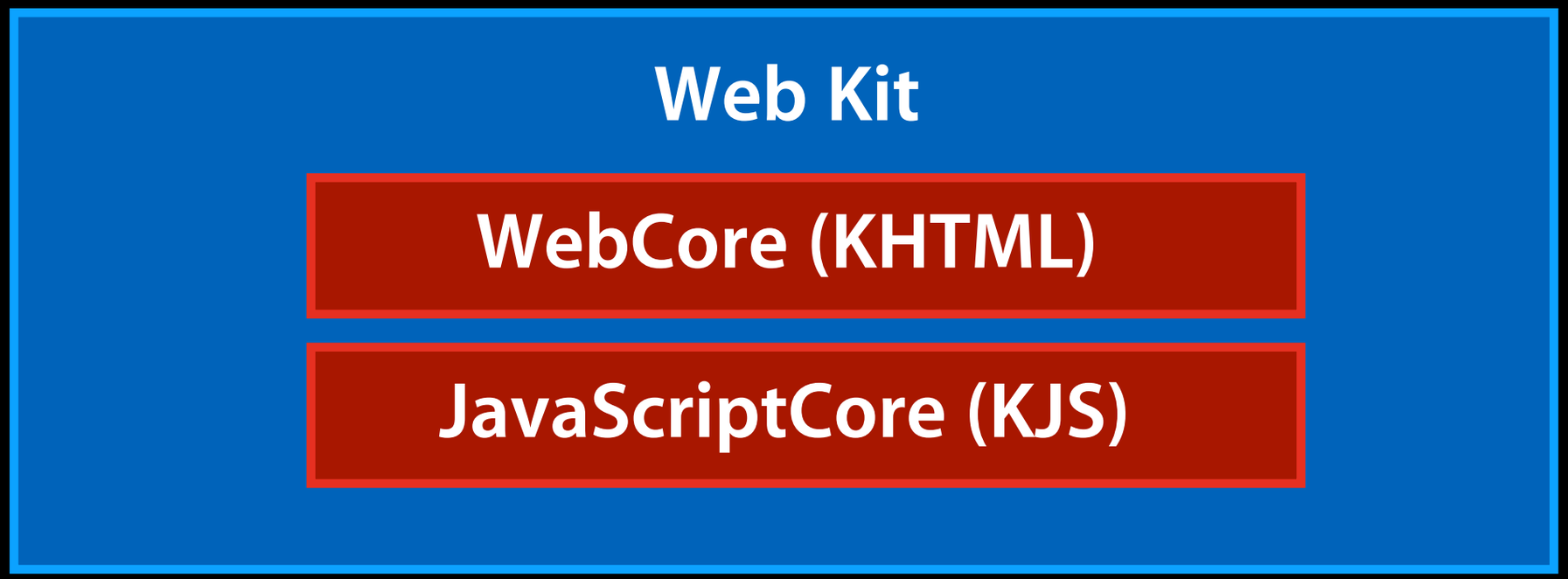
I think you're missing the reason why people are so excited about Safari.

It just works.

Mike Davidson, ESPN.com

Open Source

The Web Kit and Open Source



WebCore

- WebCore framework includes:
  - KHTML library
  - KWQ library
- WebCore is open source
- KHTML is a library from KDE, with many improvements
- KWQ is glue that makes KHTML work well on Mac OS X

JavaScriptCore

- JavaScriptCore framework includes:
  - KJS library
  - PCRE regular expression library
  - Other support libraries (all open source)
- JavaScriptCore is open source
- WebCore uses JavaScriptCore to interpret JavaScript

Why KHTML and KJS?

- We looked at many alternatives
- Many reasons for our choice, but these are the big ones:
  - Easy to modify
  - Fast, easy to make faster
  - Small

Collaborating on KHTML and KJS

- Apple continues to enhance KHTML and KJS
- KDE developers continue to enhance KHTML and KJS
- Apple gives them our enhancements
- They give us theirs

A Few of the Apple Contributions

- Hundreds of site compatibility improvements
- Performance enhancements
  - KJS speed increase 10x, even better for benchmarks
- Whitespace handling and line breaking enhancements
- Improved handling of positioned elements
- Collapsing margins
- Overflow: auto and overflow: scroll

Let’s Be Clear

- WebCore and JavaScriptCore are open source
  - You’re free to use them as the license dictates
- Should you use them?
  - ✅ Yes, via the Web Kit
  - ❌ Do not link to them directly (no API)
- Many benefits and much of the performance tuning is available only via the Web Kit

The Future

We Eat Our Own Dog Food

Sherlock

Mail

Help Viewer

Safari

XCode

Software Update

Java

We're Serious

- Apple's focus on these technologies is not just a fad
  - Expect more from Safari
  - Expect more from the Web Kit
  - Expect more enhancements to networking APIs
- We're a W3C member
  - Expect Apple to be part of web standards futures

You Should Be Too

✅ Adopt the Web Kit

❌ No reason to roll your own

The Web Kit's Advantages for You

- The Web Kit is easy to embed
- The Web Kit is fast
- The Web Kit supports standards
- The Web Kit supports real-world websites
- Safari drives improvements to the Web Kit
- The Web Kit is built in

Wrap Up

Roadmap

102 Open Source at Apple	The Mission Tue., 10:30am
403 Internet Technologies: Safari Overview	The Presidio Tue., 2:00pm
418 Internet Technologies: Advanced Foundation URL APIs	Nob Hill Thu., 3:30pm
420 Internet Technologies: Advanced Web Kit APIs	The Mission Fri., 9:00am
FF 015 Internet and Networking Technologies	The Mission Day, 10:30am

Who to Contact

John Geleynse
Manager, Software Evangelism
geleynse@apple.com

http://developer.apple.com/wwdc2003/urls.html

For More Information

Reference Library

Documentation

Displaying Web Content
ADC Home > Documentation > Cocoa > User Experience > Displaying Web Content >

Web Kit Reference for Objective-C
ADC Home > Documentation > Cocoa >

Web Kit Reference for C
ADC Home > Documentation > Carbon > HIToolbox >

URL Loading System
ADC Home > Documentation > Cocoa > Networking >

Reference Library

Apple Websites

Apple Internet Developer Website
developer.apple.com/internet

Safari Resources for Web Developers
developer.apple.com/internet/safari.html

Apple Developer Connection for Web Kit SDK
connect.apple.com

Reference Library

Books

“HTML and XHTML: The Definitive Guide”
Published by O’Reilly & Associates

“Cascading Style Sheets: The Definitive Guide”
Published by O’Reilly & Associates

“JavaScript: The Definitive Guide”
Published by O’Reilly & Associates

“Dynamic HTML: The Definitive Reference (2nd ed.)”
Published by O’Reilly & Associates

Reference Library

Standards Websites

World Wide Web Consortium

www.w3.org
validator.w3.org
www.w3.org/QA/2002/07/WebAgency-Requirements

The Web Standards Project

www.webstandards.org

“The Business Benefits of Web Standards”

devedge.netscape.com/viewsource/2003/why-web-standards

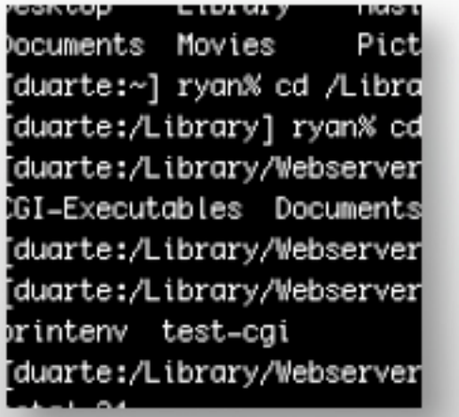
Reference Library

Standards Experts' Websites

Eric Meyer, recommended for CSS tips
diveintomark.org/safari

Peter-Paul Koch, recommended for DOM tips
www.xs4all.nl/~ppk/js

Q&A



http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference